AD-A095 451   YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE   F/G 5/7

MEMORY, MEANING, AND SYNTAX.(U)

NOV 80   R C SCHANK, L BIRNBAUM   N00014-75-C-1111

UNCLASSIFIED   TR-189   NL

1 OF 1

AD A
095451

END

DATE
FILMED

3-81

DTIC

LEVEL 12

AD A095451

FEB 25 1981

MEMORY, MEANING, AND SYNTAX

Roger C. Schank and Lawrence Birnbaum

Research Report #189

November 1980

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

MEMORY, MEANING, AND SYNTAX

Roger C. Schank and Lawrence Birnbaum

Research Report #189

November 1980

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract N00014-75-C-1111; and in part by the National Scie:e Foundation under Contract IST7918463.

To appear in "Cognitive Philosophical Computati nal Foundations of Language", T. Bever and L. Miller (Eds), in press.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) 407 051

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) RR-189	2. GOVT ACCESSION NO. AD-A095 451	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) MEMORY, MEANING, AND SYNTAX.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report.
		6. PERFORMING ORG. REPORT NUMBER (15)
7. AUTHOR(s) (10) Roger Schank and Lawrence Birnbaum		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-1111, IST7918463 NSF-IST79-18463
9. PERFORMING ORGANIZATION NAME AND ADDRESS Yale University - Department of Computer Science 10 Hillhouse Avenue New Haven, Connecticut 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE (11) November 1980
		13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited. (12) 72

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Computational Linguistics
Natural Language Processing
Syntax and Semantics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

407051

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

-- OFFICIAL DISTIRUBTION LIST --

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 copies
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 copies
Dr. Judith Daly Advanced Research Projects Agency Cybernetics Technology Office 1400 Wilson Boulevard Arlington, Virginia 22209	3 copies
Office of Naval Research Branch Office - Boston 495 Summer Street Boston, Massachusetts 02210	1 copy
Office of Naval Research Branch Office - Chicago 536 South Clark Street Chicago, Illinois 60615	1 copy
Office of Naval Research Branch Office - Pasadena 1030 East Green Street Pasadena, California 91106	1 copy
Mr. Steven Wong New York Area Office 715 Broadway - 5th Floor New York, New York 10003	1 copy
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6 copies
Dr. A.L. Slafkosky Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1 copy
Office of Naval Research Code 455 Arlington, Virginia 22217	1 copy

Office of Naval Research Code 458 Arlington, Virginia 22217	1 copy
Naval Electronics Laboratory Center Advanced Software Technology Division Code 5200 San Diego, California 92152	1 copy
Mr. E.H. Gleissner Naval Ship Research and Development Computation and Mathematics Department Bethesda, Maryland 20084	1 copy
Captain Grace M. Hopper NAICOM/MIS Planning Board Office of the Chief of Naval Operations Washington, D.C. 20350	1 copy
Dr. Robert Engelmore Advanced Research Project Agency Information Processing Techniques 1400 Wilson Boulevard Arlington, Virginia 22209	2 copies
Professor Omar Wing Columbia University in the City of New York Department of Electrical Engineering and Computer Science New York, New York 10027	1 copy
Office of Naval Research Assistant Chief for Technology Code 200 Arlington, Virginia 22217	1 copy
Major J.P. Pennell Headquarters, Marine Corp. (Attn: Code CCA-40) Washington, D.C. 20380	1 copy
Computer Systems Management, Inc. 1300 Wilson Boulevard, Suite 102 Arlington, Virginia 22209	5 copies
Ms. Robin Dillard Naval Ocean Systems Center C2 Information Processing Branch (Code 8242) 271 Catalina Boulevard San Diego, California 92152	1 copy

ABSTRACT

This paper explores the role of syntax in computational theories of natural language. We discuss the integrated processing hypothesis, which contends that meaning and world knowledge play a crucial part in language understanding even at the earliest points in the process. The hypothesis implies that syntactic knowledge plays no privileged role in language processing. Computer models of language analysis are discussed in relation to the overall theory.

MEMORY, MEANING, AND SYNTAX

Roger Schank and Lawrence Birnbaum

Yale University

1.0 Introduction

An effort has begun in the last several years to unite the common interests of those in artificial intelligence, cognitive psychology, linguistics, philosophy, and related academic disciplines. In each of these fields some subset of the community is concerned with how the human mind processes language. Thus, in some sense they all look towards a psychological theory of language that is adequate for their purposes. The emerging discipline of cognitive science, supposedly carved from the intersection of the above-mentioned fields, is thus vitally concerned with constructing such a theory of language. But, cognitive science is a discipline without discipline. There are no universally accepted methodologies, no accepted styles of explanation, and most problematic of all, there is no common language, through which a theory of language acceptable to one of the disciplines could be found acceptable to the others.

Hence, when we ask, as the organizers of this volume have asked, whether a linguistic theory is possible without a cognitive theory, or whether a cognitive theory implies a theory of language behavior, we find ourselves in a difficult situation. We all know what all the words mean, but we can be fairly sure that our view on these questions is not what another's view might be. If cognitive science is ever to make progress, however, then we had better try to start someplace.

From the perspective of artificial intelligence (AI), it is unlikely that there is, or could be, a purely linguistic theory that would be in any sense adequate. By a purely linguistic theory, we here mean a theory created to account solely for linguistic phenomena. The presupposition behind the attempt to create such a theory is that language is in some way isolable from other elements of thought. In contrast, our successes and failures in trying to construct computational models capable of performing significant linguistic tasks seem to point in another direction. Language and thought are inextricably bound together.

Part of the problem here is defining exactly what counts as a purely linguistic phenomenon. In the mid-sixties, the relevant linguistic phenomena were typically taken to be the syntactic phenomena, and hence linguistic theory was essentially syntactic theory. The assumption was that there could be a "theory of syntax".

At the root of this assumption was the competence-performance distinction (Chomsky (1965)). Without this distinction, the attempt to construct a "theory of syntax" would have seemed quite problematic (which is not to say that the distinction is itself unproblematic). No one can seriously propose that people, before they begin to decode the meaning of what they hear, must (or even can) perform a complete syntactic analysis. Such a proposal seems dubious simply on the grounds of introspective evidence. We can stop in the middle of understanding a sentence, at any point, and discover that we have already understood quite a bit of this only partially completed sentence, and that we have generated a great many expectations, based on meaning and world knowledge, concerning what might be contained in the remainder of the sentence. Clearly, we are not waiting for a complete syntactic analysis before we begin to process for meaning. More importantly, we are not relying solely on linguistic knowledge in our processing. To see this, consider the following sentence fragment:

(1) Seven Libyan gunmen shot their way into....

Our processing expectations here come from many sources: linguistic knowledge of course, but also facts about Libya, what gunmen are, the goals a Libyan gunman might have, where these gunmen might be going, and so on. We are fairly certain that they are not going into "a paper bag", for example, although that would be syntactically correct. We know that they have some goal, probably political, that it may involve money, and so on. If we next heard the word "bank" here we would not reflect on whether the "river bank" sense had been intended.

The point of all this is that no purely linguistic theory can itself explain much of this language processing behavior. The only way one can motivate a purely linguistic theory at all is to make it a competence theory. Such a theory need not pay any attention to how people process language. It can be restricted to questions of form and structure, ignoring issues of meaning, knowledge, and context.

So, is an adequate theory of language processing behavior possible without a purely linguistic theory? Of course! If a purely linguistic theory is taken to be a "theory of syntax", then our question boils down to the role of syntax within an overall theory of language and cognition.

1.1 Methodological issues

In most of our work on natural language processing, we have preferred not to use the terminology invented for the purposes of descriptive linguistics when discussing processing issues, in order to avoid possible misunderstandings. For that reason, we have avoided the use of such traditional terms as "grammar", "syntax", "semantics", and "pragmatics", in describing our theories of processing. We have instead employed terms like "conceptual" (by which we mean both semantic and pragmatic) and "inferential memory". Unfortunately, this has sometimes resulted in the kind of misunderstanding we were seeking to avoid. In particular, some cognitive scientists, on the basis of a rather cursory acquaintance with our research, have erroneously concluded that our models of language processing do not make use of syntactic knowledge. Since our models most certainly do employ such knowledge, it is worth considering their implications for syntax and its place in a cognitive theory.

There is no question that the relationship between semantics and syntax is one of the most confusing issues in cognitive science. It also seems to be a rather durable issue, since it appears over and over again in the literature on psychology, linguistics, and the philosophy of language. So, for the sake of trying to unconfuse the muddle, we will attempt to address the issue here from the point of view of the fourth discipline that comprises cognitive science, namely artificial intelligence.

What can AI contribute to the understanding of this problematic relationship? Its most important contributions, we believe, stem from its concern with theories of processing. Within the framework of artificial intelligence, we can only come to understand the relationship between memory, meaning, and syntax by understanding how and when each is utilized in the processes of understanding and generating language. Note that we do not claim that a computer model is necessarily of direct relevance to human cognitive facilities. But it is important to mention that this is true of any formal model, whether or not those who devised it employed a computer. Any formal model is ultimately subject to empirical test to determine both its psychological plausibility, and its ability to actually characterize the phenomena that it was intended to characterize. However, in the construction of our computational models, we naturally try to take into consideration that which is known about the human mechanism.

The methodology of artificial intelligence is to formulate process models, and experiment with computer implementations of those models. Interestingly enough, computer modelling has proven useful in understanding the human mechanism. One reason for this is that the attempt to implement a proposed model typically uncovers many crucial problems, the importance of which may have been grossly underestimated, or which may even have been completely unforeseen at the start. The problems raised in dealing with linguistic ambiguity provide a good example of this. When examined closely, utterances are typically quite ambiguous, yet people generally have no difficulty understanding the unambiguous message underlying an utterance in a given context. The problem of recovering a sufficiently unambiguous message from ambiguous language, of somehow evading or resolving ambiguity, is crucial to the success of any natural language processing system, be it human or machine. The centrality of this problem becomes quite inescapable the moment one tries actually to implement a model of linguistic processing. While ambiguity has been an issue in linguistics, the concern has typically been with simply being able to represent potential ambiguities. The problem of doing something about them has not been an issue.

Another reason to expect that artificial intelligence theories can contribute to our understanding of the relationship between memory, semantics, and syntax is that they must, of necessity, address the issue if they are to lead to process models capable of performing the types of linguistic tasks that people can perform. There is clearly no such methodological pressure to address the issue in other kinds of linguistic theorizing. Research aimed towards elucidating a competence theory of syntax, for example, quite naturally starts by de-emphasizing the relationships between meaning and syntax. This methodological decision has in turn often led to psychological claims for various forms of "autonomous syntax", as most clearly seen in the work of Chomsky and others of the interpretive school. These claims have often been made without consideration of the kind of relationship between semantics and syntax that might be required in order to actually perform significant language processing tasks. This lack of attention to processing considerations is quite simply impossible in artificial intelligence research.

1.2 The integrated processing hypothesis

What do our current theories say about the roles of memory, meaning, and syntax in language processing? Our working hypothesis is that meaning and world knowledge are often crucial even at the earliest points in the language understanding process (Schank, Tesler, and Weber (1970); Schank (1972); Schank (1975); Riesbeck and Schank (1976); Schank, Lebowitz, and Birnbaum (1980)). We call this theoretical position the <u>integrated processing hypothesis</u>. As we shall see, this hypothesis stands in direct opposition to theories which posit a logically autonomous syntactic analysis procedure, temporally prior to, and providing input for, semantic processing (see e.g., Fodor, Bever, and Garrett (1974); Woods (1970); Marcus (1979)). This paper is concerned with exploring the implications of the integrated processing hypothesis, the evidence for it, and its relation to other theoretical frameworks for language processing.

Two observations led us to adopt the hypothesis originally. The first was the failure of syntax-oriented approaches to the construction of natural language processing systems, particularly in the early period of machine translation research (see e.g., Bar-Hillel (1960)). This failure was due primarily to problems of ambiguity and implicit content. As we pointed out above, communication normally proceeds using language that is ambiguous and that leaves much unsaid, and these characteristics are related in that considerations of meaning and context are crucial to the solution of both problems. Thus, the failure of the syntax-oriented projects seemed to indicate that a more semantics-oriented approach to language analysis was necessary.

The second observation was the rather common-sense one that it is easier to understand a foreign language, especially when reading, than it is to speak or write it. Most of us have had the experience of picking up a magazine written in a language of slight acquaintance to us, and more or less understanding it, especially if we know something about the topic being discussed. Paraphrasing or answering questions in that language, however, would be beyond our capabilities. These facts seem to indicate that understanding language requires far less knowledge of syntax than generation, and hence that a semantics-oriented approach to language analysis could be successful.

The above considerations led to the development of a series of language analyzers based on the the low-level semantics captured by conceptual dependency theory (Schank et al. (1970); Riesbeck (1975); Riesbeck and Schank (1976); Gershman (1977) and (1979); Birnbaum and Selfridge (1980)). These analyzers proved moderately successful in a variety of settings, including story understanding, question answering, and dialog systems. At the same time, work progressed on the problem of trying to characterize the pragmatic knowledge necessary for language understanding. One of the first, and certainly the simplest, of the memory structures devised to capture this high-level semantic knowledge was the script (Schank and Abelson (1977)). A script is a temporally and causally linked set of low-level concepts describing a time-ordered stereotypical event sequence. The paradigmatic example of a scripted activity is going to a restaurant. The script notion was used as the basis of the SAM system (Schank et al. (1975); Cullingford (1978)), a computer program for understanding simple stories.

The development of these higher-level memory structures led naturally to the question of how they might be used in the language analysis process directly. The conceptual dependency parsers (see above references) were finished analyzing a sentence when they found a meaning representation for it. Thus, we had created a two-step process (language to meaning, then meaning to memory processes), that, while more sensible than a multi-step process employing an autonomous syntactic analyzer (language to syntactic structure, then syntactic structure to meaning, and so on), was still clearly unrealistic. Our first attempt to design an integrated understanding system, that is, one that used high level knowledge to help with low level parsing, resulted in a program, FRUMP (DeJong (1977) and (1979)), which applied simplified scripts directly to the problem of skimming and summarizing newspaper stories. While the task of skimming a story is not nearly as complex as deep understanding, FRUMP's success at this task reinforced our belief that the way to solve the problems of language analysis is to bring as much knowledge as possible to bear in the understanding process.

1.3 The common-sense argument

Another motivation for the integrated processing hypothesis stems from a common-sense appraisal of the role of language analysis in the language understanding process as a whole. But this raises the question, what does it mean to understand? We need not be talking exclusively about language when we talk about understanding. As understanders, we understand situations as well as sentences. We are able to operate in the world because we understand the world as well as understanding sentences about that world. In both cases, the knowledge we apply is the same. In both cases what we mean by understanding is the same.

To take a simple example from our favorite domain, when a person enters a Burger King, after having been to a McDonald's, but never before having been to a Burger King, he is confronted with a new situation which he must attempt to "understand". We can say that a person has understood such a situation (i.e., he understands Burger King in the sense of being able to operate in it) when he says "I see, Burger King is just like McDonald's."

To put this another way, we might expect that at some point during his Burger King trip he might be "reminded" of McDonald's. Understanding an experience or situation crucially involves being reminded of the previously experienced situation stored in memory that is most like the input, and being able to use that prior phenomenon as a source of expectations relevant to current processing (Schank (1979) and (1980)).

When we are reminded of some event or experience in the course of undergoing a different experience, we are reminded of that experience because the structures we are using to process that new experience are the same structures we are using to organize memory. Thus, we cannot help but pass through the old memories while processing a new input. There are an extremely large number of such high level memory structures. Finding the right structure out of all those available (that is, the one that is most applicable to the experience at hand) is what we mean by understanding.

In this view then, understanding some input involves finding the most relevant higher-level structure available to explain the input, and creating a new memory structure for the input that is derived from the old structure. Understanding is a process that has its basis in memory then, particularly memory for closely related experiences accessible through reminding and expressible through analogy.

But what does such a view have to say about how we understand language? We are claiming that people understand by utilizing expectations that come from the memory structure that has the most in common with the situation currently being processed. Our point is that these expectations need not be solely about events, they are about language as well. Viewed this way, understanding language means accessing the most relevant memory to help process the situation being described. Expectations about language are not necessarily different than those about situations. In fact, they are likely to be tightly bound to each other. A language understander need not, for example, explicitly choose among all the possible meanings of "demand" in a terrorism story. What we know about terrorism colors our expectations about what will happen in that situation, and about what the words that describe that situation mean. Understanding language depends heavily on the knowledge we have about the situations that language describes. The common-sense motivation for integrated processing, then, is that it doesn't make sense <u>not</u> to make use of this relevant information in processing further input.

An important use of this information, as we just pointed out, is in somehow evading or resolving ambiguity. Consider the following example:

(2) The old man's glasses were filled with sherry.

Most people incorrectly decide that "glasses" means "eyeglasses" in the course of processing the above sentence, and so are surprised by the outcome: they must consciously "back up". This error can only be explained by assuming that people apply their knowledge of the relation between age and eyesight when analyzing this sentence, <u>before</u> any putative low-level analysis of the sentence is complete.[1] If "bartender" is substituted for "old man" in (2), the opposite assumption is made in processing, and so no "back up" occurs. These sorts of assumptions are made while a sentence is being processed. To do that, world knowledge must be accessed and applied while everything else is going on.

2.0 <u>What is the problem?</u>

The issue of how memory, meaning, and syntax are related in language processing can be divided into three distinct (though closely related) questions. These questions are often conflated, and we believe that the failure to keep the distinctions clear has been a

----

[1] An interesting point about (2) is that the feeling of surprise experienced is quite similar to that produced by a sentence like

(3) The horse raced past the barn fell.

This apparent similarity provides an anecdotal basis for believing that the failure in a first pass reading of both (2) and (3) above is in fact the failure of the same process in both cases, though in one case because of pragmatic factors, and in the other case syntactic factors.

major factor inhibiting mutual comprehension among those who have argued the issue. Therefore, before we discuss exactly what the integrated processing hypothesis claims, and its implications for theories of processing, we will first try to clear up some of the misunderstanding.

When discussing some computational process, it generally proves useful (in models of cognitive processes at least) to distinguish three aspects of such a process. The first aspect of interest is the <u>control structure</u> of the process, which defines the task to be accomplished (i.e., what is input and what is output), the sub-processes which must be invoked in order to accomplish the task, how they communicate, and the order in which they must be invoked. The second aspect is the <u>representational structures</u> which are constructed and operated on by the process, and which constitute the outputs of any intermediate stages and of the process as a whole. The final aspect is the <u>knowledge base</u> on which the process draws to actually perform the task. Often, this knowledge is in the form of rules, and so the knowledge base is typically called a rule base. To distinguish these aspects is not to claim that they are realized by distinct elements in the process: it is simply a heuristic to aid in the analysis of the process. Our three questions, then, concern what integration, as opposed to autonomy, of syntax might possibly mean for these three aspects of a process.

(1) The first of the three questions concerns the processes which apply the knowledge used in language understanding: Are conceptual (semantic and pragmatic) knowledge and syntactic knowledge applied by

separate control mechanisms, or is there an integrated control structure which applies both? That is, does language understanding proceed as a unified process, or as several separate processes?

(2) The second question involves representational structures. Are the structures which are used to encode semantic or conceptual information in memory separate from those which are used to encode syntactic information, i.e., is there an independent level of syntactic representation? What kinds of representations are computed by people as part of the understanding process?

(3) The final question concerns the knowledge used in understanding. Can the rule base which embodies semantic and world knowledge be separated from that which embodies syntactic knowledge? Is there a clean separation between these sets of rules or is there a continuum of rules, some purely semantic, some purely syntactic, some in between?

While it is important to distinguish these questions, the possible answers to them are of course interdependent. We will see this in what follows.

2.1 Control structures

The question of whether syntax and semantics are applied by a unified control mechanism or by independent mechanisms is of course the question of whether or not an autonomous syntactic analysis procedure exists. The claim that a single control mechanism applies both is the weakest possible claim of the integrated processing hypothesis. That's because if this _isn't_ true, it's quite difficult to argue for integration along any of the other dimensions of the process. On the other side, the claim that syntax and semantics are applied by independent control mechanisms is the strongest possible form of autonomous syntax.

It would be useful at this point to map out some of the possible alternatives concerning the issue of integrated versus autonomous control processes for applying memory, semantics, and syntax to the problem of language analysis. What follows are sketchy descriptions of several possible positions:

[a] Semantics and syntax are completely separable. According to this position, syntactic analysis is a completely independent process, logically and temporally prior to the meaning-based inference processes involved in understanding. This position implies that syntax alone controls the analysis at the earlier points of processing. This is the view that results from a straightforward if naive recasting into the performance domain of the model outlined in Chomsky (1965) and descendent models.

[b] Semantics and syntax are "nearly decomposable". In this view, there is still a prior syntactic analysis process, the output of which provides the input for semantic processing. However, this process may on occasion query a semantic component in order to make a syntactic decision. This limited interaction between semantics-oriented processes and the syntactic analysis process is controlled by syntax, in that the decision that some interaction is required can be taken only by the syntactic mechanism. This is the position taken by Fodor et al. (1974) with their theory of

independent syntactic processing within clauses, and by Woods (1970) and Marcus (1979) with somewhat more flexible communication regimes between the syntactic and semantic components.

[c] Semantics and syntax have a "heterarchical" relationship. According to this position, semantics-oriented knowledge and syntactic knowledge are still applied by separate control processes. However, their relationship is far more cooperative than in the previous position, with the two processes operating in a manner similar to coroutines. That is, the interaction is no longer exclusively under the control of the syntactic mechanism. A syntactic component does some work, then calls some semantic process which does what it can and then in turn calls syntax for more information, and so on. This appears to be the position advocated by Winograd (1972) and (1977).[2]

[d] Semantics and syntax are employed in an integrated control structure. In this view, the decision as to whether to utilize syntactic knowledge or conceptual knowledge is made by a single control structure, and whatever available knowledge is most useful will be applied in trying to analyze and understand the input. This is the position that we have taken (Schank et al. (1970); Riesbeck and Schank (1976); Schank et al. (1980)). A similar view seems to inform the experimental program being carried out by Marslen-Wilson and his colleagues (see e.g., Marslen-Wilson, Tyler, and Seidenberg

----

[2] This description might in fact apply to models which also fit the previous or subsequent descriptions, since "independence" is a fuzzy concept: as the richness and frequency of communication between modules increases, they become more integrated and less independent.

(1978)). One way in which the totally integrated view might be differentiated from the heterarchical view relates to the question of syntactic representations; Winograd's model constructs and operates on them, while the models to be presented here do not. We will return to this point in the next section.

Note that the opposition on the above spectrum of positions between [a], logically separate syntactic analysis, and [d], integrated control structure, is often misconstrued in the following way. Since position [a] implies that syntax alone controls early language analysis, the opposite of that must be something like "semantics alone controls early language analysis". This is clearly wrong, but that in no way affects the validity of position [d], which doesn't imply anything of the sort.

2.2 Representational structures

Any language processing system must compute some structures for representing meaning, on the one hand, and words and their properties, on the other, in the course of understanding or generating language. An important question which guides our investigation of language processing theories is: What additional structures must be computed to represent the syntactic information associated with utterances? There are basically two positions that one can take:

[a] An autonomous level of syntactic representation (e.g., phrase markers) must be computed. For example, Fodor et al. (1974) claim that "the structural analyses to be recovered are ... precisely the trees that a grammar generates," (p. 368), by which they mean that

syntactic structures of the sort postulated by generative linguistics are computed by people in the course of language comprehension.

[b] No independent level of syntactic representation is constructed or operated on during language processing. This claim has an important consequence for our theories of language: whatever syntactic distinctions need to be represented, must be represented either at the level of conceptual structures, or at the level of words. This can be interpreted in two ways. Straightforwardly, it could mean that the syntactic representation is "part of" the conceptual representation, in the sense of being part of the same data structure, indexed with it, etc., but nevertheless serving a distinct function. On a more radical interpretation, this can be taken as asserting that if a conceptual representation carries syntactic information which is necessary for subsequent processing, this information must also serve some semantic or pragmatic function. In our own models, then, whenever we add elements to a representation in memory for the purpose of carrying syntactic information, we must try to justify them independently in terms of some conceptual function.[3]

----

[3] This point, interestingly enough, bears a mirror-image resemblance to an argument put forward by Katz and Postal (1964), in support of the hypothesis that syntactic transformations must preserve meaning. As they pointed out, this claim implies that any difference in meaning between two sentences must be reflected in some difference in the syntactic deep structures underlying those sentences. Further, they recognized that in order to support the original hypothesis, their job as theorists was to justify such deep structure differences on independent syntactic grounds.

The kinds of representational structures that are computed has implications for whether an integrated control structure performs both syntactic and semantic processing, or, instead, they are accomplished by independent control structures. The reason is that one of the characteristics that would lead us to say that one process was independent of another, would be if it acted on, and produced as output, different sorts of structures. Hence, without the computation of independent syntactic representations as part of the language understanding process, one of the characteristics that might lead us to single out some independent syntactic processor would be missing. In this sense, the computation of an independent level of syntactic representation is a weak prerequisite for the existence of an independent syntactic processor. Hence, claims of independent syntactic processing are usually accompanied by claims that independent syntactic representations are needed. And so, in arguing against an independent syntactic processor, our claim of an integrated conceptual and syntactic processor is accompanied by the additional claim that conceptual and syntactic representations are integrated as well.

2.3 Knowledge base

The strongest possible claim that an integrated processing hypothesis could make would be that the knowledge used for understanding is totally integrated. This would mean that all rules used in language processing reference at least some semantic information, and hence that no purely syntactic rules exist. On the other side, the claim that some purely syntactic rules exist is the weakest possible claim of autonomous syntax. Without such a set of autonomous syntactic rules, for example, the claim that there is an autonomous syntactic processor doesn't even make sense. That is, the extent to which the rules which embody syntactic knowledge can be separated from the rules which embody semantic knowledge will determine whether a logically prior or separate syntactic analysis procedure is possible or sensible.

We do not advance the claim that no purely syntactic rules exist, hence we do not support the strongest possible form of the integrated processing hypothesis. However, even in a weaker form the hypothesis does make an interesting claim about the knowledge used in understanding: While there most likely are some exclusively syntactic processing rules, these rules simply occupy one extreme of a continuum of rules, and are not distinguished <u>by use</u> from other sorts of rules. This position follows from the two prior claims of the integrated processing hypothesis that (1) language processing is effected by an integrated control process, and (2) there is no independent level of syntactic representation computed in language processing. If these two claims are true, then whatever "exclusively syntactic" might mean, it does not mean <u>functionally</u> distinguishable in use from other sorts of rules. Thus, the integrated processing hypothesis is supported to the extent that purely syntactic rules can be shown to play a similar role in processing to other kinds of rules. We will turn to that after we have presented our process models in some detail.

2.4 Integrated processing revisited

We are now in a position to state exactly what the integrated processing hypothesis claims. First, it claims that language analysis proceeds as a unitary process, integrating all kinds of knowledge, rather than as a collection of separate processes, one for each kind of knowledge. Again, this is in contrast to the models proposed by Fodor et al. (1974), Woods (1970), and Marcus (1979). Second, it claims that no independent level of syntactic representation is being constructed, operated on, or output by the language analysis process. This is in contrast to all of the above models, as well as the model proposed by Winograd (1972). Third, it claims that, although there are rules which are in some sense purely syntactic, such rules are not *used any* differently than other sorts of rules, i.e. they are functionally integrated in processing and play no privileged role. This follows from the first two claims.

3.0 Psychological evidence

There has been a great deal of psychological experimentation that bears on the syntax-semantics relationship in language analysis. In this section, we will review a few of these results that seem to support the integrated processing hypothesis. Among psychologists, the results of this work have convinced even the strongest partisans of generative linguistics of the following two facts:

[a] There is no evidence that people make use, in comprehension or generation, of the kinds of rules devised by generative linguists to describe linguistic phenomena.

[b] The very strong claim of a completely autonomous syntactic processor (position [a] in section 2.1) cannot be upheld.

In fact, points [a] and [b] above constitute the most "conservative" interpretation of the experimental results, in the sense of conserving <u>some</u> role for generative linguistics in psychology. Less sympathetic observers will note that the results, while consistent with various patched-up syntactic autonomy claims, were not as predicted by theorists who advocate that position.

One of the earliest, and most significant, results was uncovered by Slobin (1966). He investigated differences in how long it takes to understand passive sentences as compared to active forms, distinguishing between reversible and non-reversible passives. A reversible passive is a sentence like "John was seen by Bill," in which syntax must be inspected to determine who saw whom. That is, the only way to distinguish this from the sentence "John saw Bill," is by noticing that one is passive construction, and the other is not, because both are equally sensible. A non-reversible passive is a sentence like "The ice cream cone was eaten by John," in which by virtue of semantics one can determine who ate what. That is, this sentence can be distinguished from "The ice cream cone ate John," on the grounds that the latter does not make much sense. Slobin found that, although reversible passives take longer to understand than active forms, non-reversible passives <u>do not</u>. This result indicates that, at the very least, semantic grounds are used to determine whether or not some syntactic evidence should be checked. What is even more damaging to models like Fodor et al. (1974) or Marcus

(1979), is that if the syntactic evidence to be checked consists of some underlying syntactic phrase marker which must be computed, these results raise questions as to whether that computation would be made at all if other kinds of evidence obviated the need for it. Either way, the result is completely congenial to the integrated processing hypothesis.

Marslen-Wilson and his colleagues have done numerous studies concerning the status of autonomous syntactic processing models. A representative result can be found in Tyler and Marslen-Wilson (1977). They studied a model proposed by Fodor et al. (1974), a chief claim of which was that, within clauses, sentence analysis proceeded by the operation of a completely autonomous syntactic processor, and no higher-level knowledge could enter the process until a clause boundary was reached. Subjects were presented with sentence fragments like:

(1) If you walk too near the runway, landing planes...

(2) If you've been trained as a pilot, landing planes...

and then immediately supplied with either the probe word "is" or "are". On pragmatic grounds, as determined by the content of the first clause, "is" is appropriate as a continuation of (2) but not (1), while "are" is appropriate for (1) but not (2). Subjects were asked to simply repeat the probe word as quickly as possible. The data showed that they were slower to repeat an inappropriate probe. The only way to determine whether or not a probe was appropriate was on the basis of meaning and pragmatic knowledge, making use of the context created by the content of the initial clause of the test sentence fragments. Since the appropriateness of the probe was a syntactic property (number agreement), and since subjects were probed in the middle of an uncompleted clause, this result demonstrates that whatever syntactic processing is going on is not independent of meaning-oriented processing, even within clauses.

In more recent work, Shwartz (1980) examined several possible low-level strategies for determining pronominal referents, many of which had previously been proposed in the literature. The utilization of some of these proposed strategies depended on the existence of explicit syntactic representations. The study found no evidence for the use of such strategies. This kind of result is important, because to the extent that processes which might have been thought to depend on explicit syntactic representations can be found not to, the integrated processing hypothesis is strengthened.

One final study we will mention concerns an investigation into the putative independence of semantic and pragmatic processing in language understanding. Gibbs (1979) investigated a claim by Clark and Lucy (1975), among others, that understanding indirect speech acts requires computing, in a fairly bottom-up fashion, the "literal meaning" of the utterance, which is then used as input to special pragmatic interpretation rules which discover the "real meaning". Clark and Lucy had shown that, in the absence of any context, comprehension of indirect speech acts did take longer than, e.g., direct requests. This was taken as evidence that an extra processing step was being performed, presumably involving the application of the special pragmatic rules to the previously computed "literal meaning" of the input. Gibbs performed a similar study, however this time embedding the indirect speech acts in a suitable context. He found that, in context, the interpretation of indirect speech acts takes no longer than that of direct language, thus calling into question the claim that "literal meaning" must be computed.

4.0 Process models of language analysis

The problem of language analysis is, given some linguistic input in some context, to determine the semantic and memory structures underlying that input. The goal of conceptual analysis is to perform this task in a manner consistent with the integrated processing hypothesis, namely as a unified process, directly in one step. This goal differs rather drastically from that assumed by most models of language analysis, which hold that the process employs a parsing mechanism which performs an explicit syntactic analysis of the input sentences.[4] In this section, we will describe several approaches to conceptual analysis, involving various kinds of semantic and world knowledge in the parsing process. Most of these have been implemented in running computer programs. We will also discuss some more speculative approaches which have not been implemented.

----

[4] Descriptions of syntactic parsers can be found in Thorne, Bratley, and Dewar (1968); Bobrow and Fraser (1969); Woods (1970); Winograd (1972); Kaplan (1975); Marcus (1975) and (1979). Other examples of semantics-oriented analyzers are those described by Wilks (1973), (1975), and (1976); Burton (1976); Rieger and Small (1979); Wilensky and Arens (1980).

4.1 Expectation-based analysis

A language understander must connect concepts which are obtained from word meanings, and from inferences derived from word meanings, into a coherent representation of the input as a whole. Because of the possibilities of word-sense ambiguity or irrelevant inference, an understander must also be able to choose among alternative concepts. So, conceptual analysis consists mainly of connecting and disambiguating (i.e., choosing among) conceptual structures.

The processing knowledge which a conceptual analyzer uses for this task is in the form of expectations (Schank et al. (1970); Riesbeck (1975); Riesbeck and Schank (1976)). When a person hears or reads words with certain meanings, he expects or predicts that words with certain other meanings may follow, or may have already been seen. People constantly predict what they are likely to see next, based on what they have read and understood so far and on what they know about language and the world. They use these expectations to disambiguate and connect incoming text.

Of course, expectations are used in syntactic analysis programs as well. The difference lies in the origin of the expectations. In syntactic analyzers, the expectations are derived from a grammar. In conceptual analyzers, the expectations are governed instead by the incomplete conceptual structures representing the meaning of the input.

We will illustrate the use of expectations in understanding by way of an extremely simple example. Suppose that the following sentence were the input to an expectation-based conceptual analysis system: "Fred ate an apple." Reading from left to right, the system first finds the word "Fred". The system understands this as a reference to some male human being named Fred, and stores the reference, represented as the token FRED, in some kind of short-term memory. The next word is "ate". This is understood as an instance of the concept of eating, which in conceptual dependency (Schank (1975)) is represented by a case frame something like this:

(INGEST ACTOR (NIL) OBJECT (NIL)).

Also, the meaning of "ate" supplies some expectations which give hints as to how to go about filling the empty slots of this frame. One of these expectations suggests that the ACTOR of the INGEST concept, an animate being, may have already been mentioned. So the analyzer checks short-term memory, finds FRED there, and fills the ACTOR slot of the INGEST:

(INGEST ACTOR (FRED) OBJECT (NIL)).

There remains an unfulfilled expectation, which suggests that some mention will be made of what it is that Fred ate, that it should be some edible thing (or at least a physical object), and that it should fill the OBJECT slot. Next, the word "an" is read. This creates an expectation that some instance of a concept should follow, with the instruction that if one is found, it should be marked as an indefinite reference. (This is information which can aid memory.)

Finally, "apple" is read. It is understood as an instance of the concept APPLE, representing something which is known to be food. The expectation created when "an" was read is satisfied, so APPLE is marked as an object not previously seen, which we will represent as: (APPLE REF (INDEF)). The second expectation created when "ate" was read is also satisfied, so the OBJECT slot of the INGEST is filled by (APPLE REF (INDEF)). The system's current understanding of the input is represented as

(INGEST ACTOR (FRED) OBJECT (APPLE REF (INDEF))).

There are no more words to read, so the process halts.

This simple example gives the flavor of the conceptual analysis process.[5] When a word is read, the conceptual structure representing the meaning of that word is added to short-term memory. In addition, expectations (governed by the unfilled case slots) are created, which provide the processing knowledge necessary to connect up these conceptual structures into a representation of the input as a whole. Connections are established between concepts via the filling of empty case slots.

Several analyzers based on these ideas have been implemented. Given our basic assumption about the need to understand within a context, we cannot illustrate the parsing process directly here without putting it within the context of other processes. Below we

----

[5] For more complete, technical descriptions of this process, see Riesbeck (1975); Riesbeck and Schank (1976); Gershman (1979); Birnbaum and Selfridge (1980).

have an example of output from the CYRUS system. CYRUS is a program that automatically builds up a data base from news stories analyzed by the FRUMP program mentioned earlier. It can then be queried in natural English and it will respond in English. The major point behind CYRUS is its ability to reconstruct input questions so that they conform to how information is stored in its memory. The CYRUS memory model (Schank and Kolodner (1979)) employs one of our analysis programs, called CA, to analyze questions. What follows is an example of a question answering session with CYRUS. (CYRUS was written by Janet Kolodner. CA was written by Lawrence Birnbaum and extended by Martin Korsin.)

```
***************

@CYRUS

Are you interested in Muskie or Vance?  (M or V) : *VANCE


*(DIALOG2)

Enter next question
>When was the last time you were in Egypt?

The question is:
((ACTOR HUM1 IS (*LOC* VAL POL6)) TIME TIME0)
The question type is "time"
The question concept is:
((ACTOR HUM1 IS (*LOC* VAL POL6)))
   inferring a diplomatic trip
answering question using time context
 directly
The answer is:
(CON113)

 on DEC 10 1978.

Enter next question
>Why did you go there?

The question is:
((CON (*?*) REASON
       ((ACTOR HUM1 <=> (*PTRANS*) OBJECT HUM1 TO POL6))))
The question type is "motivational"
```

```
The question concept is:
((ACTOR HUM1 <=> (*PTRANS*) OBJECT HUM1 TO POL6))
answering question using previous context:
 directly
The answer is:
(CON145)

 to negotiate the Camp David Accords.

Enter next question
>Who did you talk to there?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) TO (*?*)) PLACE POL6 TIME TIME8)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) TO (*?*)) PLACE POL6 TIME TIME8)
   inferring undifferentiated political meeting
answering question using previous context:
 directly
The answer is:
(CON184 CON198)

 with Anwar Sadat.

Enter next question
>Has your wife ever met Mrs Begin?

The question is:
((ACTOR HUM4 IS (*PROX* VAL HUM60W)) TIME TIME15 MODE (*?*))
The question type is "verification"
The question concept is:
((ACTOR HUM4 IS (*PROX* VAL HUM60W)) TIME TIME15)
   inferring social occasion
searching memory for question concept
  searching directly for input -- sM-SOCIAL-OCC
   found (GN590)
The answer is:
(CON233)

 yes, most recently at a state dinner in Israel in JAN 1980.

Enter next question
>What heads of state have you met?

The question is:
((ACTOR HUM1 IS (*PROX* VAL TOKQ14)) TIME TIME22)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 IS (*PROX* VAL TOKQ14)) TIME TIME22)
   inferring a diplomatic meeting
searching memory for question concept
  searching directly for input -- $MEET
   ...
applying strategies to search memory
```

```
  ...
   searching for sM-CONFERENCE
      found (GN505 GN492 GN420)
    ...
   searching for sM-VIPVISIT
      found (CON95)
    searching sM-VIPVISIT instance for input
    ...
   searching for I-NEGOTIATE
      found (GN545A GN462 GN468A GN420A GN404A)
    ...

 King Hassan, Menachim Begin, Mustafa Khalil, Ian Smith,
 President Assad, King Hussein, and Anwar Sadat.

Enter next question
>Last time you went to Saudi Arabia, where did you stay?

The question is:
((<=> ($SLEEP ACTOR HUM1 LOC (*?*))) TIME G0682)
The question type is "concept completion"
The question concept is:
((<=> ($SLEEP ACTOR HUM1 LOC (*?*))))
   inferring a diplomatic trip
answering question using time context
 directly
The answer is:
(G0715)

 in a palace in Saudi Arabia on SEP 23 1978.

>Did you go sightseeing there?

The question is:
((<=> (sM-SIGHTSEE ACTOR HUM1))
      MODE (*?*) TIME G0784 PLACE POL11)
The question type is "verification"
The question concept is:
((<=> (sM-SIGHTSEE ACTOR HUM1)) TIME G0784 PLACE POL11)
answering question using previous context:
 directly
 using top-level events
The answer is:
(G0822)

 yes, at an oilfield in Dharan on SEP 23 1978.

***************
```

How consistent is this kind of conceptual analysis process with the integrated processing hypothesis? One failing should be clear immediately: the above discussion has touched only on the role of low-level semantics and conceptual representations in parsing. Higher-level memory structures have not been integrated. On the other hand, the model is consistent with the hypothesis in that it constructs a conceptual representation directly, using a unified control structure, and without prior syntactic analysis. Further, no independent level of syntactic representation is built or operated on.

4.2 The role of syntax

What is the role of syntax in a conceptual analyzer? Traditional notions of syntax use categories like "part of speech" and "phrase marker" in discussing the structure of a sentence. What we would like to claim in this section is that these notions of syntax are inappropriate when attempting to describe and utilize syntactic knowledge in a language understanding process.

To begin with, what is the purpose of syntactic knowledge? Clearly, a major use of syntactic knowledge is to direct the combination of word meanings into utterance meaning when semantic information is not sufficient or is misleading. For example, in an utterance like "Put the magazine on the plate," it is syntactic knowledge that tells the understander which object is to be placed on top of which. From the point of view of its use in a conceptual analyzer, therefore, a large part of syntax is knowledge of how to combine word meanings based on their positions in the utterance. This knowledge is necessary whenever there are several gaps in a representational structure which have the same semantic requirements. For example, both the ACTOR and the TO slots of an ATRANS (concept of transfer of an abstract property) can appropriately be filled by a "higher animate". Syntactic knowledge must then be used to decide which of several appropriate gaps a structure should fill.

How can this syntactic knowledge be characterized? We seek a specification which takes into account the fact that the point of syntax is its <u>use</u> in the understanding process. We have viewed part of the process of understanding as one of connecting representational structures, where a connection has been established between structures when one fills a slot in the other, or both fill a larger form. Thus syntactic knowledge is knowledge which specifies where in the utterance some word is to be found whose meaning can be connected (via slot-filling) with the meaning of another word. Of course, we must now specify the notion "position in an utterance".

Given that processing knowledge is encoded in expectations, this problem reduces to the question of how positional information can be taken into account in those expectations. This may be done by having tests that check for:

Relative positions of concepts in short-term memory.

The proper order for filling slots in structures.

These methods describe position in an utterance by the use of relative positional information. Both are essentially ways of utilizing word order information. The first of these methods, using relative position in short-term memory, describes the position of a conceptual structure in direct relation to the other structures it might be connected with via slot-filling.

The second method, ordering of the slots to be filled, relates the position of a structure to other structures somewhat more directly. In particular, rather than utilizing temporal order of input as reflected in the order of concepts in short-term memory, it can be directly employed by constraining the order in which slots should be filled, For instance, a constraint that the ACTOR slot of a conceptualization be filled before the OBJECT slot reflects the fact that the ACTOR of that conceptualization should be seen before the OBJECT in an active construction.

Another important method of utilizing syntax in a conceptual analyzer is by relating the position of a conceptual structure to the position of a particular lexical item (i.e., a function word), rather than to another concept. Function words, including prepositions, post-positions, affixes, complementizers, and so forth, are very important syntactic cues in conceptual language analysis. For example, in a sentence like

John gave Mary to the Shah of Iran.

the function word "to" clearly marks the recipient of the action, in a case where semantics alone is simply not sufficient.

4.3 Functional integration of syntactic rules

One of the crucial features of the analysis model we have been discussing is the <u>functional</u> integration of syntax and semantics. Purely semantics-based rules, purely syntax-based rules, and mixed rules all play the same role in processing: they are being used to construct, connect, and disambiguate conceptual structures.

We can illustrate this with a few examples. The conception of syntax sketched out in the last section is that syntactic knowledge will be used when semantics alone is insufficient to determine the meaning of the input. One common cause of such a situation, we pointed out, is that the semantic restrictions on case slots in a conceptual representation are not always unique (i.e., are not mutually exclusive). This suggests that in those cases where some case slot <u>does</u> have unique semantic requirements, no syntactic knowledge would be necessary to find the correct filler for that slot. Hence, that filler should have an extremely free syntax with respect to the entire construction.

It turns out that one can find examples of this sort of phenomenon. Consider the conceptual object of MTRANS (concepts of communication). Concepts of communication take an entire conceptualization, or proposition, as their object, namely that concept which is being communicated. Since, on semantic grounds, no other case role of an MTRANS can be filled with a complete proposition, we would expect that the relative position of the object concept with respect to the overall MTRANS construction would vary rather freely, and it does. Consider the following examples:

(1a) A Liberian tanker ran aground off Nantucket Island, the Coast Guard said.

(1b) The Coast Guard said a Liberian tanker ran aground off Nantucket Island.

(1c) A Liberian tanker, the Coast Guard said, ran aground off Nantucket Island.

These examples suggest that the rule (expectation) which is used to fill the object slot of MTRANS conceptualizations is completely semantics-based. It simply looks for an entire conceptualization to fill the object slot.

On the other hand, relative subclauses, possibly initiated by the word "that", as in

(2) The car that I saw in the showroom...

require a great deal of syntactic knowledge to be properly analyzed. "That" sets up several expectations, one of which, rendered in English, says roughly:

> To the right will be found a concept with some unfilled slot(s). Use the concept to the left to fill (one of) the slot(s), in accordance with semantic requirements. Then take the resulting conceptualization and subordinate it to the concept on the left.

This rule is purely syntactic: it refers to relative positional information, and to "unfilled slots", but it says nothing about the kind of concepts or restrictions on slots. This is not to say that the rule can be utilized without reference to semantics, however, since semantic knowledge must be consulted to produce a meaningful subordinate conceptualization. The important point is that this purely syntactic rule plays exactly the same role in processing as does the purely semantic rule for finding the object of an MTRANS. Syntactic and semantic knowledge cannot be distinguished by use in this model of language analysis.

This same point can be made by considering generation, for instance of noun groups. A purely syntactic rule that one might want to have in a generator is that adjectives precede nouns. But, the order of the adjectives themselves is not determined by such a purely syntactic rule. A generator must have enough knowledge to know that "big red ball" is generally more appropriate than "red big ball", and that "old Irish grandmother" is more appropriate than "Irish old grandmother". Examples like this can best be explained by postulating the rule that an adjective which supplies information about a more intrinsic property of the modified noun will be closer to that noun than adjectives which describe less intrinsic properties (Clark and Clark (1977)). The proper generation of noun groups depends crucially on the simultaneous application of both this rule, and the purely syntactic rule that adjectives precede nouns. But the notion of "intrinsic property" is clearly a conceptual, not a syntactic, notion. Thus, the problem of generating proper adjective order is another example which argues for the functional integration of purely syntactic rules with other sorts of rules.

4.4 Using higher-level memory

So far, we have only discussed the integration of low-level semantics, of the sort characterized by conceptual dependency, into the analysis process. However, the success of the FRUMP system in using higher-level memory structures early on in the parsing process showed that, in addition to low-level semantics, they too must play a role in language analysis.

This "integrated understanding" approach enables the analysis of language directly into the high level representations which organize memory and facilitate understanding. Often it is an entire high level memory structure which best represents the meaning of some input. The sentence "Fred proposed to Wilma," for example, should call to mind a large structure representing our knowledge of courting, proposals, and marriage.

In addition to directly invoking these higher-level structures, the parsing process should try to fill the empty roles in these structures, just as in the case of lower level conceptual frames. The key point here is that we would like an analysis process which can fill slots in higher-level structures directly, in one step. The best way to explain the distinction is with an example. Suppose a language understander is reading the input

(3) A plane carrying federal marshals...

Given that the analyzer can discover the relevant high level structure (call it $AIRPLANE), we would like the analyzer to fill the PASSENGERS slot of that structure with "federal marshals" directly, rather than first using them in some intermediate, lower level structure, and then subsequently transferring the information up to the PASSENGERS slot of $AIRPLANE.

The attempt to integrate higher-level knowledge in a somewhat deeper understander than FRUMP resulted in the Integrated Partial Parser, or IPP (Schank et al. (1980); Lebowitz (1980)). One of the key ideas underlying this approach to language analysis is that the direct application of high level memory structures can focus the analyzer's attention, controlling what should be pursued, and what should be ignored. This is possible because high level structures typically supply very specific expectations.

To understand how this system works, consider the following example, taken from the New York Times:

> An Arabic speaking gunman shot his way into the Iraqi Embassy here (Paris) yesterday morning, held hostages throughout most of the day before surrendering to French policemen and then was shot by Iraqi security officials as he was led away by the French officers.

As IPP processes each word from the input, one of three things can occur. The word can be completely ignored, it can be saved in short-term memory and then skipped, or it can be completely processed immediately.

The first possibility is that it may simply be skipped. There are many words which have no significant conceptual content for the level of analysis which IPP strives to model. Examples from the above story include the words "most", "way", and "held".

The second possibility is that a word may be saved in the analyzer's short-term memory, and then skipped. Words for which this processing strategy seems appropriate have some conceptual content, but of a rather dull and uninteresting sort. Nevertheless, we cannot simply ignore them, because their meanings may be important in elaborating our knowledge of the events or things that we <u>are</u> interested in. For example, they may be used to fill roles in the conceptual structures representing interesting events. In many cases, however, they will never be used again. Examples include the words "Arabic", "Iraqi", and "his".

Two things can ultimately be done with these words. Either their meaning does help elaborate something interesting, in which case that meaning will be incorporated in the final representation, or it doesn't. For example, the meaning of the word "police" in the phrase

(4a) before surrendering to French police

is incorporated into the representation because we are interested in whom the terrorist surrendered to. On the other hand, the meaning of the word "officers" in the phrase

(4b) as he was led away by French officers

is not incorporated into the meaning representation because it does not add to our knowledge of anything interesting.

The conceptual content of these words will often have some associated processing information, in the form of expectations, which can help elaborate on their meaning. For example, the concept underlying "embassy" has an expectation which looks for the name of the country which the embassy represents, and the concepts underlying the words "police", "officers", and "officials" have expectations for the name of the governmental authority in whose name they operate. But if a word is subject to a "save and skip" strategy, these expectations should not be applied until we know that the concept that they would strive to elaborate itself elaborates on our knowledge of something interesting. This is because if it turns out that we don't care about that concept, we don't want to have done unnecessary work in elaborating it. Let's compare our processing of "police" in phrase (4a) above with our processing of "officers" in phrase (4b). Since it turns out that the concept of police in the first phrase adds to our knowledge of an interesting event, the expectation associated with the concept that looks for the governing authority should actually be used. Hence, the modifier "French" would be added to the representation. In the second case, since the concept of "officers" does not add to our knowledge of anything interesting, there is simply no point in applying any similar rule. In this case, the modifier "French" would be ignored.

The third possible processing strategy we can apply to a word is to process it immediately, i.e. pay attention to its meaning and the expectations that it generates. This is the strategy that we apply when the word has significant and interesting conceptual content. It is these concepts and their associated expectations that drive the analysis. Examples from the above story include the words "gunman", "shot", and "hostages". The expectations which these words generate include the same kind of simple elaborative, or "slot-filling", expectations associated with some of the words for which a "save and skip" strategy is appropriate. For example, the word "gunman" generates an expectation looking for the nationality or political affiliation of the gunman.

These words can also generate expectations which operate at a much higher level. For example, when we read the word "gunman", we expect to read that he may have performed the action of shooting a weapon. We also expect that the events associated with several possible scripts, including $ROBBERY and $TERRORISM. These expectations operate in a manner somewhat akin to script application (Cullingford (1978)) in that they serve to recognize events, and so recognize that they are sensible in the given context. So, once we know that the gunman is quite likely a terrorist, we expect that he may hold hostages, that he may shoot or kill some people, and that he may make some demands. We also know that there are only a small number of possible outcomes of the episode: the terrorist might be captured, he might surrender, he might be killed, or he might escape. These high level expectations help us decide what is important in the text in a very top-down way. The analysis process depends crucially on this. But its flexibility also depends on its ability to pursue questions about interesting things and events, even if they were not anticipated.

Below is an example of the input/output behavior of IPP in processing the above story. (IPP was written by Michael Lebowitz.)

```
***************

Input:

(AN ARABIC SPEAKING GUNMAN SHOT HIS WAY INTO THE IRAQI
 EMBASSY HERE THIS MORNING HELD HOSTAGES THROUGHOUT MOST OF
 THE DAY BEFORE SURRENDERING TO FRENCH POLICEMEN AND THEN
 WAS SHOT BY IRAQI SECURITY OFFICIALS AS HE WAS LED AWAY BY
 FRENCH OFFICERS)


Output:

** MAIN EVENT **                   ** UNEXPECTED EVENTS **
EV1 =                              EV4 =
 SCRIPT    $TERRORISM               ACTION    PROPEL
 ACTOR     ARAB GUNMAN              ACTOR     IRAQI OFFICIALS
 PLACE     IRAQI EMBASSY            OBJECT    ARAB GUNMAN
 INTEREST  9.                       ITEM      *BULLETS*
 CITY      PARIS                    DIR-FR    GUN
 TIME      MORNING                  INTEREST  5.
 SCENES                             AFTER     EV3
    EV2 =                           RESULT
     SCENE     $HOSTAGES               EV6 =
     PLACE     IRAQI EMBASSY             STATE     DEAD
     ACTOR     ARAB GUNMAN               ACTOR     ARAB GUNMAN
     INTEREST  7.                        INTEREST  4.
     TIME      DAY
    EV3 =
     SCENE     $CAPTURE
     PLACE     IRAQI EMBASSY
     OBJECT    ARAB GUNMAN
     ACTOR     POLICEMEN
```

```
INTEREST  6.
AFTER     EV2
```

673. msec CPU (0. msec GC), 944. msec clock, 1759. conses

****

IPP produces this conceptual representation of the input in an integrated fashion[6], employing a detailed knowledge base concerning terrorism in order to process the story in a top-down fashion. The program applies what it knows about the situation being described to help it process individual words. In this story, for example, it did not treat "held" as an ambiguous word. Rather, it was expecting the possibility of reading about the taking of hostages at that point in the story and it simply skipped "held" to see what the next word was. This view of processing is crucial for dealing with the many English words which are virtually meaningless, such as "hold", "take", "have", "get", and so on.

4.5 The problem of flexibility

In the last section, we presented the idea of analyzing linguistic input directly into the memory structures which encode pragmatic information. However, the process of directly filling slots in these structures with concepts from the input text immediately raises the problem of integrating conceptual and syntactic knowledge in a flexible way. This problem arises because knowing where in a sentence to look for some slot-filler is obviously syntactic

----

[6] Exactly how it did this is beyond the scope of this paper. More details can be found in Schank et al. (1980) and Lebowitz (1980).

knowledge, while knowing which slot in a memory structure the concept actually fills is obviously conceptual knowledge. How are these different types of knowledge put together?

In order to understand why flexibility is an issue, consider the following example. When we read the sentence fragment

(5) A plane carrying...

we know that what comes next in the sentence is most likely the passengers or cargo. Having expectations about the position of some information in a sentence is, quite clearly, syntactic knowledge. Yet we know that what comes next in the sentence should be used to fill the either the PASSENGERS or CARGO slot of the $AIRPLANE memory structure. This is clearly conceptual information, and of a rather high level at that.

A first solution to the problem of how a language analyzer can put these two types of knowledge together is to make both available in the high-level memory structures. IPP follows this approach, and it has proven quite successful. It seems quite plausible that the word "demand" and its various linguistic properties should be stored with the memory structure representing knowledge of cerrorism, or that the word "order" and its properties should be stored with the memory structure for restaurants.

On the other hand, it seems far less plausible that the $AIRPLANE memory structure has detailed knowledge of every construction in the English language that can be used to describe a change of location, and that is what would be needed. For, instead of the word "carrying", we could have substituted a phrase like "in which they were flying..." or innumerable other phrases.

Another possible solution is to store all the knowledge at the lexical level. That is, the expectations arising from lexical items describing a change of location would know how to fill the PASSENGERS slot whenever the $AIRPLANE context were active. Again, it seems implausible that every construction in the English language which can describe a change of location contains some words which know about what to do when operating in the context of the $AIRPLANE structure. Thus, while either of these two approaches might well be used in certain circumstances, taken to an extreme they would lead to an obvious and damaging lack of extensibility across domains.

Before proposing a solution to this problem of somehow flexibly integrating knowledge from different levels[7], we have to confront another, closely related problem. What should the representation of "A plane carrying federal marshals..." look like? Even if the word "carry" had some general PTRANS (transfer of location) sense, the representation of this sentence fragment should not be a low-level conceptual structure as

(6) (PTRANS ACTOR (PLANE0) OBJECT (MARSHALLS0))

From the considerations we've already discussed, it seems clear that the representation should reference the high level $AIRPLANE memory

----

[7] What follows is somewhat speculative. The processes sketched out in the remainder of this section have been implemented in part in the BORIS story understanding system (see Lehnert, Dyer, Harley, Johnson, and Yang (1980)).

structure:

(7) ($AIRPLANE AIRCRAFT (PLANEO) PASSENGERS (MARSHALLSO))

But, this representation can't come from the word "carried". It has to come from "plane". On the other hand, "carried" contributes something to the meaning, since if we substituted the phrase "mowed down" the meaning would be entirely different. How can we accommodate these two necessities in one process?

Suppose, for the moment, that "carried" has a meaning which is simply a sort of generalized PTRANS. Additionally, let's suppose that the following structures (among others) are in memory:

```
(8) plane (lexical item)
        \
         \associated concept
          \
           *AIRPLANE*
                \
                 \associated script
                  \
                   ($AIRPLANE
                     AIRCRAFT (*AIRPLANE*)
                     PASSENGERS (?X)
                     ....      )
                         |
                         |main conceptualization
                         |
                   (PTRANS OBJECT (?X) ....)

    carry (lexical item)
        \
         \associated concept
          \
           (PTRANS ACTOR (NIL) OBJECT (NIL) ....)
```

Notice that there is a match between the generalized PTRANS meaning of "carry", and the PTRANS which is the main conceptualization (see Cullingford (1978)) of the memory structure $AIRPLANE. Now, suppose a language analyzer had the following general rule:

> If a slot-filler of a concept has an associated higher memory structure, the main conceptualization of which matches that concept, then that associated memory structure is the actual meaning, and should be substituted for the concept.

Using a rule like this, "A plane carried ..." would parse into an instance of $AIRPLANE. Furthermore, this use of memory structures enables a more flexible solution to the problem of merging processing knowledge from different levels. In the course of performing the kind of pattern matching that this rule needs, the OBJECT of the PTRANS which came from "carry" would be identified as the PASSENGERS slot of $AIRPLANE. So, any lower-level, somewhat syntactic expectation (perhaps associated with the word "carry") which tries to fill the OBJECT slot of the PTRANS could then be used to <u>directly</u> fill the PASSENGERS slot of $AIRPLANE.

In order to understand how this works, let's examine the memory structures above. Notice that the variable ?X is used as a place-holder in both the PASSENGERS slot of the $AIRPLANE structure, and in the OBJECT slot of its main conceptualization, which is a PTRANS. This is how a memory structure can represent the fact that those two slots should be filled by the same token. Now, when the analyzer matches the PTRANS from "carry" with the PTRANS which is the main conceptualization of $AIRPLANE, it will realizes that, of course, the OBJECTs of the two PTRANSs must be the same. Since the OBJECT (?X) of the PTRANS which is the main conceptualization of $AIRPLANE is the same as the PASSENGERS of that structure, by transitivity the OBJECT of the PTRANS arising from the word "carry" is the same as the

PASSENGERS of the $AIRPLANE. In other words, the lower-level expectation from "carry" which seeks to fill the OBJECT slot of the PTRANS, would automatically fill the PASSENGERS slot of $AIRPLANE without ever having to know about it. And conversely, $AIRPLANE fills its PASSENGERS slot without having to know the syntax of "carrying". In general, all the lower level rules for filling slots in the PTRANS can be applied to directly filling the corresponding higher level slots of $FLY. Using a scheme like this, it is possible to merge the high level, very specific semantic restrictions on slot fillers, with low-level syntactic restrictions.

Now, consider text fragments like: "A train carrying federal marshals...", "A bus carrying federal marshals...", etc. In each of these cases, the associated memory structure $TRAIN or $BUS is the correct representation. And in each of these cases, the process described above would allow the lower level expectations associated with "carry" to directly fill the correct slots in these scripts.

These examples show how this method might enable the flexible integration of high level conceptual knowledge and lower-level syntactic knowledge in language analysis. The lower level expectations derived from lexical items do not need to know every single high level structure to which they might be applicable. And the higher-level structures do not need to know about all of the possible English constructions which indicate where the filler of some high-level slot might be in a sentence. But through this kind of process, a language analyzer can still parse directly into high level memory structures.

5.0 A look at memory

We have been talking a great deal about parser-memory interaction, but generally from the point of view of parsing: we have said little about the nature of memory. In Schank and Abelson (1977) we postulated some memory entities that we labeled scripts, plans, goals, and themes. These entities were responsible for supplying expectations about what happens in a situation, thus facilitating the inference process by applying real world knowledge, for example, of human habits and motivations.

Since then we have revised our theories somewhat, attempting to make them both more general and more consistent with the available psychological evidence. For example, Bower, Black, and Turner (1979) noticed that people were confused between events that happened in similar scripts. This led to the realization that scripts which shared common elements had to be making use of some memory structure that held those common elements. Further revisions stemmed from our attempt to somehow account for the phenomenon of being reminded of particular experiences (Schank (1979) and (1980)).

This revised view of memory makes use of an entity that we call a Memory Organization Packet (or MOP). While it is certainly beyond the scope of this paper to present a detailed view of MOPs, it is instructive to take at least a cursory look at their role in processing to see how they might relate to what we have been saying here about expectation and its role in the parsing process.

As we mentioned near the beginning of this paper, the most important aspect of memory is the role it plays in facilitating understanding. It seems obvious that we cannot really understand anything without relying at least to some extent upon the information that we have stored in memory. But it may not be as obvious that any structure that we postulate as a memory structure must, by its very nature, also be a processing structure, as well as an indexing structure for specific memories.

Scripts had been originally conceived as static data structures that contained stereotypical information about standard situations such as restaurants, airplane rides, doctor visits, and so on. Psychological evidence for the use of scripts by people makes it clear that such entities must exist (see e.g., Bower et al. (1979), Abelson (1980)). But people do not store information about stereotypical situations apart from information about times in their lives that their expectations went awry. We remember funny circumstances. Further, we tend to remember these circumstances at precisely the moment when similar funny things happen. When we are asked to write down our order in a restaurant, we tend to recall (be reminded of) a previous time when we were asked to do the same thing. This kind of reminding is so ubiquitous, that it seems unlikely that it would turn out to be epiphenomenal.

Since people cannot know beforehand under what circumstances they will want to recall a particular memory, it seems clear that they must store unusual memories in terms of the memory structure from which they deviate. That is, memories must be stored in terms of the processing structures that these memories related to. The reason to do this seems clear enough. How are we to learn from the failure of an expectation unless we store such a case with the failed expectation itself? Since people can generalize from similar situations, they must be capable of storing unusual situations in a place where subsequent situations, unusual in a similar way, would be likely to bring them to mind. Expectation failures could then be the basis of generalization and learning processes.[8]

If scripts are not simply static structures to be used by some "script applier mechanism", then the demands on them change. Information shared by any two scripts should be a significant generalization which is derived from them, and subsequently is stored "outside" them in memory, at a single location. To do this requires ascertaining what it might mean for two scripts to share the same information, and finding out when such sharing is actually "realized" in memory and when it is not.

While we require that general information be stored in only one place in memory, specific episodes are treated differently. Specific episodes can be multiply categorized, that is, remembered as instances of many different phenomena at once. Thus, experiences can be recalled through many different aspects. Often, pointing to an episode by way of a particular characterization of that episode can cause other parts of that episode to be recalled. In those instances the different aspects are being used as pointers to the one place in

---

[8] This subject is explored further in Schank (in press).

memory where the entire episode resides.

The place where similar memories reside we call a MOP. The purpose of a MOP is to provide expectations that enable the prediction of future events on the basis of previously encountered, structurally similar events. These predictions can be at any level of generality or specificity. The creation of a suitable MOP provides a class of predictions organized around the common theme of that MOP. The more MOPs that are relevant to a given input, the more predictions will be available to help in understanding that input and the better the understanding will be. MOPs must be able to make useful predictions in somewhat novel situations for which there are no specific expectations, but for which there are relevant experiences from which generalized information is available.

To see how MOPs function in processing, we will consider the information relevant to a visit to a doctor's office. The primary job of a MOP in processing new inputs is the creation of expectations about what will happen in that situation. At least five MOP's help to provide the processing structures necessary for understanding a doctor's office visit. They are: PROFESSIONAL-OFFICE-VISIT; MAKE-CONTRACT; FIND-SERVICE-PROFESSIONAL; USE-SERVICE; and FIX-PROBLEM. As we will see, these five MOPs overlap quite a bit. There is nothing wrong with that, indeed it should be expected that any memory theory would propose overlapping structures, since they are the source of memory confusions and enable the construction of useful generalizations across domains.

The primary function of the PROFESSIONAL-OFFICE-VISIT MOP (henceforth POV) is to provide the correct sequencing of the scenes appropriate to a visit to the given type of professional. POV is not the only MOP active in processing a story about a visit to a doctor's office, however. So, while we focus on POV in the following example, it should be kept in mind that the problem of knowing which MOPs are active and when is still relevant.

Consider the following story:

(1) I went to the doctor's yesterday. While I was reading a magazine I noticed that a patient who arrived after me was being taken ahead of me. I am going to get even by not paying my bill for six months!

In processing the first sentence of this story, the first problem is to call in the relevant MOPs insofar as we can determine them. The phrase "went to the doctor" cannot be processed simply by "summing up" the meanings of its constituent words. The phrase must invoke what we know about doctor's offices, sick people, paying bills, and so on. What we need to do is find a place in memory that will tell us that this knowledge may be relevant for processing this story. That is, our first problem is establishing which MOPs to access. The phrase "went to the doctor" tells us this because the lexical item "doctor" points information stored under the node DOCTOR in memory. That is, part of what we know about doctors is that, when a person goes to one, it is usually because he is sick; that he will have to visit an office; pay a bill; and so on. Thus, the relevant MOPs are activated by finding the DOCTOR node in memory and looking there to find what MOPs are activated by the PTRANS of some individual to the location of a doctor, as described in this phrase.

These MOPs are now connected together to build up a series of expectations about what might happen at the doctor's. We do this by examining the enablements and temporal precedences for each of the activated MOPs. Thus, for example, POV dominates a set of scenes that involve getting to the office, waiting in the waiting room, seeing the professional, and then taking care of the bill and possibly making another appointment, in that order. FIX-PROBLEM includes the possible strategy that a professional who specializes in the particular type of problem might be found. This in turn invokes FIND-SERVICE-PROFESSIONAL, which itself invokes USE-SERVICE and MAKE-CONTRACT. By tracking the logical and temporal enablements, we can, after having read the first sentence, assume that the speaker had a health problem; that he knew a doctor to go, or in found one in some other way; that he had the money to pay, and so on. Our contention here is that the implicit assumption of these facts by the understander is necessary to continue processing.

Certainly, the presence of these implicit assumptions can be demonstrated. For example, when information is present that contradicts these assumptions, we usually feel called upon to mention it. We might ask "How did you get the money?" or "Where did you find a doctor?" if we discover that there is likely to have been a problem with those aspects of the situation. The ability to do this depends upon having access to the relevant MOPs so that we can find the implicit assumptions that were called into question.

The expectations established by POV have the following form:

INITIATOR: MAKE-APPOINTMENT

PRECONDITION: [be there]

SEQUENCE: ENTER + WAITING ROOM + ENTER OFFICE + [get service]
+ LEAVE OFFICE + (MAKE NEW APPOINTMENT) + EXIT

FOLLOWED-BY: [go elsewhere] (after EXIT) +
[get bill] + [pay bill] (from MAKE-CONTRACT)

The items in square brackets above are empty slots that must be filled by scenes from some other MOP, since they are not unique to POV itself. Scenes in upper case indicate items unique to POV. The scenes are the structures that store actual memories and use those memories to in processing.

The slots represented by [be there] and [go elsewhere] are both normally filled by the LOCAL-TRAVEL MOP, which points to standard scripts for getting to and from specific places, such as $BUS, $CAR, $WALK, and $SUBWAY. However, since the story does not get more specific on this point, these slots will remain empty. Should it become necessary to fill these slots, they do contain information about the destination or departure point which the LOCAL-TRAVEL MOP can use in checking the plausibility of whatever script tries to fill the slot.

The more interesting gap in POV is the [get service] slot. Responsibility for filling this slot is left to the MOP DOCTOR-SERVICE, which was activated by USE-SERVICE on the basis of the kind of service desired. The standard actions of a doctor performing his service will be recognized as normal at this point in the event sequence being constructed:

```
                PROF-OFFICE-VISIT    MAKE-CONTRACT
      ___________/ /   |        \   /  /         \
     /            /    |         \/   /          PAY BILL
MAKE-APPT+[get there]+WAITRM+[get service]+[go away]
                              /    /
                          DOCTOR-SERVICE
                          /      /
                      USE-SERVICE
```

In order to use POV we had to have encoded, as part of what we know about doctors, that a visit to the doctor entails a POV. But doctors are not the only professionals whose offices are visited by clients. Any other MOP that specifies a set of events that normally take place in an office that people visit in order to avail themselves of some service, must also somehow point to POV. Our knowledge about lawyers, dentists, accountants, and so on would also include such links. This immediately raises up the question of whether or not the strands that link POV and these various other MOPs are traversable in both directions. The problem of combinatorially explosive memory search would be greatly aggravated if these strands were in fact traversable both ways. Further, the use of bi-directional links would imply that people could easily enumerate the various kinds of professionals who have such offices. It seems more likely that to perform that task, one would first employ some method to enumerate various types of professionals, and then check whether they have such offices.

So, we can conclude that POV has a large number links coming into it from the various MOPs that invoke it, and that it provides a set of scenes for those MOPs. In fact, the main job of POV is to provide an ordering relationship on a set of scenes, not all of which come from POV directly. POV is mainly used to organize, according to temporal precedence, information found elsewhere. Those scenes which are actually dominated by POV may be unique to it, or may be shared with another MOP. PAY-BILL, for example, is shared with MAKE-CONTRACT (which is also active here because of its relation to POV).

This notion of sharing can be taken quite literally. That is, different aspects of bill paying episodes will be stored in memory as part of the MOPs from which they came. WAITING-ROOM, on the other hand, will contain all the actual memories of sitting in waiting rooms because it is a scene that is not shared by any other MOP. It is this property of <u>not</u> overlapping with other MOPs that (somewhat counterintuitively) causes memory confusions. The reasoning is as follows: Since WAITING-ROOM is not shared, anything that happens in a waiting room is stored exclusively under that heading. But, since a great many MOPs use POV as their setting, they all leave some of the actual memories for episodes which they have been invoked to explain, stored under WAITING-ROOM. Since the strands which link these MOPs to POV are not bi-directional, they all use WAITING-ROOM as a resource, but do not get back any of the actual memories. These memories stay in WAITING-ROOM. This results in memory confusions about waiting room episodes between different MOPs that use POV. The actual consultation with the doctor or lawyer, however, will be stored in different MOPs, and hence not confused.

Searching through a MOP serves to identify both content strands, i.e. the actual scenes contained by the MOP, and empty strands, i.e. place holders for information contributed by some other MOP. In POV for example, WAITING-ROOM is a content strand. It contains a great deal of information, such as what waiting rooms look like, what they contain, what happens there and so on. The [get service] strand on the other hand is entirely empty. It serves to relate the rest of POV to whatever content is placed there by the MOP that invoked POV. This is where DOCTOR-SERVICE or DENTIST-SERVICE come into play. These MOPs contain only the most specific information concerning what a doctor does in fulfilling [get service] for a patient.

We can now return to the processing of our example story. As we said, the first sentence activates POV, which defines the sequence of events to be expected. When the next sentence

> (1a) While I was reading a magazine I noticed that a patient who arrived after me was being taken ahead of me.

is encountered, it can be interpreted in terms of knowledge stored WAITING-ROOM scene of POV, since that is the first (and only) scene found in POV which can normally include reading magazines. The WAITING-ROOM scene also contains pointers to the purpose it fulfills in the overall MOP, in this case, knowledge of the social convention of queueing or "waiting one's turn" to utilize a scarce resource. Interpreting the remainder of the second sentence in terms of this knowledge results in the recognition of a violation of that social convention by the doctor.

The final sentence

(1b) I am going to get even by not paying my bill for six months!

refers to the PAY-BILL scene of MAKE-CONTRACT. Here again, a violation of the social conventions for an action is described, as well as a reference to a high level structure, REVENGE, which is used to interpret the relation between the two violations.[9]

The point of this extended discussion of MOPs was to give an overview of the entire understanding process. The relation to language analysis is exemplified by the processing of a sentence like "I am going to get even by not paying my bill for six months!" The memory structures built prior to the processing of this sentence are of great use in analyzing a semantically complicated action like "get even". Knowledge of a prior motivation allows an understander to quickly understand this phrase, either without ever considering the lexical ambiguity and resulting possibilities for syntactic ambiguity, or by immediately choosing the correct meaning. Either alternative requires expectations derived from the active high level memory structures.

But, saying that, we must ask where those expectations come from, i.e. we must understand how the relevant memory structures get invoked. An expectation about bill paying can only come from having activated a structure like MAKE-CONTRACT. But this structure was

----

[9] REVENGE is not a MOP, but a related type of structure, called a TOP, which deals with rather different sorts of knowledge (see Schank (1979)).

never explicitly mentioned. Thus, in order to process efficiently we must infer the relevance of certain memory structures from the input and from what we know about the world, and use the expectations derived from those structures in the immediate processing of later input. Only an integrated memory and parser combination can accomplish this.

6.0 Conclusion

A cognitive theory of language embodies a linguistic theory: the two are inseparable. This runs counter to the notion of a purely linguistic theory as a competence theory, i.e., one that purposefully excludes the issues raised by how people actually use language.

For example, Chomsky (1965) rules out memory as an issue in linguistic theory. The notion of memory employed there, a static view of memory as simply "a place to put things", implied that its only relevance is in the limitations it imposes. Thus a competence model of language did not need to deal with memory. But a richer, more sophisticated view of memory shows that it can be an advantage. It allows us to predict the right sense of "order" when the context, situational or textual, is a restaurant. Indeed, memory can help us predict almost the entire meaning of what has yet to be said in an utterance while we are still in the middle of hearing it. These predictions are an integral part of any cognitive/linguistic theory. What a person expects about the sentence he is hearing is based on what he knows of the world, his language, and language processing. The point is that all parts of the understanding process bear upon one another. No one of them can, ultimately, be explained apart from the rest.

Idealization is an effective method in science. But the notion of linguistic competence was always much more than simply an idealization of the speaker-hearer. That is because language, particularly syntax, is just the tip of the phenomena we are studying. Language cannot be separated from the reason for its existence and use in human society. Communication is at the heart of language after all. Nevertheless, the commitment to a competence theory of syntax has led Chomsky (1980) to deny even this rather obvious point:

> Attempts to provide some sense to the notion that communication is somehow the essential property of language have not been very successful in my view. (p. 54)

In contrast, we claim that a theory of how language effects communication must be the ultimate goal of a cognitive linguistic theory. No study of language can achieve this goal if it fails to view communication as a process.

As communicators, we have two roles. We are called upon to understand the utterances of others, and we are called upon to generate our own. In performing this latter task, we must formulate the ideas we want to express and then, while we are still engaged in this formulation process, we must begin to encode those ideas into language. If the purely linguistic rules played some privileged role in this process, then there might be some motivation for a purely linguistic theory of those rules (and even then only if the rules were of a sort which could actually be utilized by the generation process). We have endeavored to show here that this is not the case, but we have barely begun to explore all the factors which affect language generation in a theoretically significant way. These must include even social context, since there are clearly generation rules which depend on that. For example, the rules that determine when to use a stock phrase like "would you be so kind" in making a request will depend on the relationship of the person making the request to the person receiving it, and on the impression he wishes to make. If a language production process is to utilize this kind of knowledge, we must face the issue of how it fits in with the other sorts of knowledge that must be employed.

What we have just said for generation is even more true of language understanding. The goal of understanding language is in reality to understand ideas. Language is a means of communicating those ideas and, as such, is merely so much baggage to be stripped away to reveal the contents. The rules we employ in this stripping process are, of course, of great interest. But, again, the purely linguistic rules play no privileged role. The knowledge we need includes knowledge of who does what in restaurants, of how to bring appropriate memories to bear when they are needed, as well as rules about where to look in a sentence for a word that is likely to satisfy one's expectation for a certain kind of object. No one type of knowledge is more crucial than any of the others in the process as a whole, and studying any type separately leaves open the most important question: how that knowledge functions within the integrated totality of the communication process. Thus, a full theory of language understanding will have to await a full theory of cognition.

generation in a theoretically significant way. These must include even social context, since there are clearly generation rules which depend on that. For example, the rules that determine when to use a stock phrase like "would you be so kind" in making a request will depend on the relationship of the person making the request to the person receiving it, and on the impression he wishes to make. If a language production process is to utilize this kind of knowledge, we must face the issue of how it fits in with the other sorts of knowledge that must be employed.

What we have just said for generation is even more true of language understanding. The goal of understanding language is in reality to understand ideas. Language is a means of communicating those ideas and, as such, is merely so much baggage to be stripped away to reveal the contents. The rules we employ in this stripping process are, of course, of great interest. But, again, the purely linguistic rules play no privileged role. The knowledge we need includes knowledge of who does what in restaurants, of how to bring appropriate memories to bear when they are needed, as well as rules about where to look in a sentence for a word that is likely to satisfy one's expectation for a certain kind of object. No one type of knowledge is more crucial than any of the others in the process as a whole, and studying any type separately leaves open the most important question: how that knowledge functions within the integrated totality of the communication process. Thus, a full theory of language understanding will have to await a full theory of cognition.

References

Abelson, R. 1980. The psychological status of the script concept. Research report no. 2, Yale University, Cognitive Science Program, New Haven, CT.

Bar-Hillel, Y. 1960. The present status of automatic translation of languages. In F. Alt, ed., Advances in Computers 1, Academic Press, New York, pp. 91-163.

Birnbaum, L., and Selfridge, M. 1980. Conceptual analysis of natural language. In R. Schank and C. Riesbeck, eds., Inside Computer Understanding, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 318-353.

Bobrow, D., and Fraser, B. 1969. An augmented state transition network analysis procedure. In Proceedings of the International Joint Conference on Artificial Intelligence, Washington, D.C., pp. 557-567.

Bower, G., Black, J., and Turner, T. 1979. Scripts in text comprehension and memory. In Cognitive Psychology, vol. 11, pp. 177-220.

Burton, R. 1976. Semantic grammar: An engineering technique for constructing natural language understanding systems. Research report no. 3453, Bolt Beranek and Newman, Inc., Cambridge, MA.

Chomsky, N. 1965. Aspects of the Theory of Syntax. The MIT Press, Cambridge, MA.

Chomsky, N. 1980. Rules and representations [including responses]. In The Behavioral and Brain Sciences, vol. 3, pp. 1-61.

Clark, H., and Clark, E. 1977. Psychology and Language. Harcourt Brace Jovanovich, Inc., New York.

Clark, H., and Lucy, P. 1975. Understanding what is meant from what is said: A study in conversationally conveyed requests. In Journal of Verbal Learning and Verbal Behavior, vol. 14, pp. 56-72.

Cullingford, R. 1978. Script application: Computer understanding of newspaper stories. Research report no. 116, Yale University, Dept. of Computer Science, New Haven, CT.

DeJong, G. 1977. Skimming newspaper stories by computer. Research report no. 104, Yale University, Dept. of Computer Science, New Haven, CT.

DeJong, G. 1979. Prediction and substantiation: A new approach to natural language processing. In Cognitive Science, vol. 3, pp. 251-273.

Fodor, J., Bever, T., and Garrett, M. 1974. The Psychology of Language. McGraw-Hill Book Co., New York.

Gershman, A. 1977. Conceptual analysis of noun groups in English. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, pp. 132-138.

Gershman, A. 1979. Knowledge-based parsing. Research report no. 156, Yale University, Dept. of Computer Science, New Haven, CT.

Gibbs, R. 1979. Contextual effects in understanding indirect requests. In Discourse Processes, vol. 2, pp. 1-10.

Kaplan, R. 1975. On process models for sentence analysis. In D. Norman and D. Rumelhart, eds., Explorations in Cognition, W. H. Freeman Co., San Francisco, CA, pp. 117-135.

Katz, J., and Postal, P. 1964. An Integrated Theory of Linguistic Descriptions. The MIT Press, Cambridge, MA.

Lebowitz, M. 1980. Generalization and memory in an integrated understanding system. Research report no. 186, Yale University, Dept. of Computer Science, New Haven, CT.

Lehnert, W., Dyer, M., Harley, S., Johnson, P., and Yang, C. 1980. BORIS: A progress report. Research report no. 188, Yale University, Dept. of Computer Science, New Haven, CT.

Marcus, M. 1975. Diagnosis as a notion of grammar. In Proceedings on Theoretical Issues in Natural Language Processing, Cambridge, MA, pp. 6-10.

Marcus, M. 1979. A theory of syntactic recognition for natural language. In P. Winston and R. Brown, eds., Artificial Intelligence: An MIT Perspective, The MIT Press, Cambridge, MA, pp. 191-229.

Marslen-Wilson, W., Tyler, L., and Seidenberg, M. 1978. Sentence processing and the clause boundary. In W. Levelt and G. Flores d'Arcais, eds., Studies in the Perception of Language, John Wiley and Sons, Chichester, U.K., pp. 219-246.

Rieger, C., and Small, S. 1979. Word expert parsing. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan, pp. 723-728.

Riesbeck, C. 1975. Conceptual analysis. In R. Schank, ed., Conceptual Information Processing, North-Holland Publishing Co., Amsterdam, pp. 83-156.

Riesbeck, C., and Schank, R. 1976. Comprehension by computer: Expectation-based analysis of sentences in context. Research report no. 78, Yale University, Dept. of Computer Science, New Haven, CT. Also appears in W. Levelt and G. Flores d'Arcais, eds., Studies in the Perception of Language, John Wiley and Sons, Chichester, U.K., 1978, pp. 247-293.

Schank, R. 1972. Conceptual dependency: A theory of natural language understanding. In Cognitive Psychology, vol. 3, pp. 552-631.

Schank, R. 1975. Conceptual Information Processing, North-Holland Publishing Co., Amsterdam.

Schank, R. 1979. Reminding and memory organization: An introduction to MOPs. Research report no. 170, Yale University, Dept. of Computer Science, New Haven, CT.

Schank, R. 1980. Language and memory. In Cognitive Science, vol. 4, pp. 243-284.

Schank, R. (in press). Failure-driven memory. In Cognition and Brain Theory.

Schank, R., and the Yale AI Project. 1975. SAM: A story understander. Research report no. 43, Yale University, Dept. of Computer Science, New Haven, CT.

Schank, R., and Abelson, R. 1977. Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, R., and Kolodner, J. 1979. Retrieving information from an episodic memory, or why computers' memories should be more like people's. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan, pp. 766-768.

Schank, R., Lebowitz, M., and Birnbaum, L. 1980. An integrated understander. In American Journal of Computational Linguistics, vol. 6, pp. 13-30.

Schank, R., Tesler, L., and Weber, S. 1970. Spinoza II: Conceptual case-based natural language analysis. Research report no. AIM-109, Stanford University, Dept. of Computer Science, Stanford, CA.

Shwartz, S. 1980. The search for pronominal referents. Unpublished ms., Yale University, Cognitive Science Program, New Haven, CT.

Slobin, D. 1966. Grammatical transformations and sentence comprehension in childhood and adulthood. In Journal of Verbal Learning and Verbal Behavior, vol. 5, pp. 219-227.

Thorne, J., Bratley, P., and Dewar, H. 1968. The syntactic analysis of English by machine. In D. Michie, ed., Machine Intelligence 3, American Elsevier Publishing Co., New York, pp. 281-309.

Tyler, L., and Marslen-Wilson, W. 1977. The on-line effects of semantic context on syntactic processing. In Journal of Verbal Learning and Verbal Behavior, vol. 16, pp. 683-692.

Wilensky, R., and Arens, Y. 1980. PHRAN: A knowledge-based natural language understander. In Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp. 117-121.

Wilks, Y. 1973. An artificial intelligence approach to machine translation. In R. Schank and K. Colby, eds., Computer Models of Thought and Language, W. H. Freeman Co., San Francisco, CA, pp. 114-151.

Wilks, Y. 1975. A preferential, pattern-seeking semantics for natural language inference. In Artificial Intelligence, vol. 6, pp. 53-74.

Wilks, Y. 1976. Parsing English I and II. In E. Charniak and Y. Wilks, eds., Computational Semantics, North-Holland Publishing Co., Amsterdam, pp. 89-100 and 155-184.

Winograd, T. 1972. Understanding Natural Language. Academic Press, New York.

Winograd, T. 1977. On some contested suppositions of generative linguistics about the scientific study of language. In Cognition, vol. 5, pp. 151-179.

Woods, W. 1970. Transition network grammars for natural language analysis. In Communications of the ACM, vol. 13, pp. 591-606.

DA

FIL

3 -